

# 實驗 1-認識 Arduino 的 I<sup>2</sup>C 通訊功能和利用 RFID

## 建立門禁管理系統

### † 學習目標：

1. 學習 Arduino Uno 與其開發環境使用方法。
2. 學習實體 Arduino 的數位和類比控制。
3. 認識 I<sup>2</sup>C 通訊功能
4. 學習 RFID 模組 RC522 與蜂鳴器控制方法。

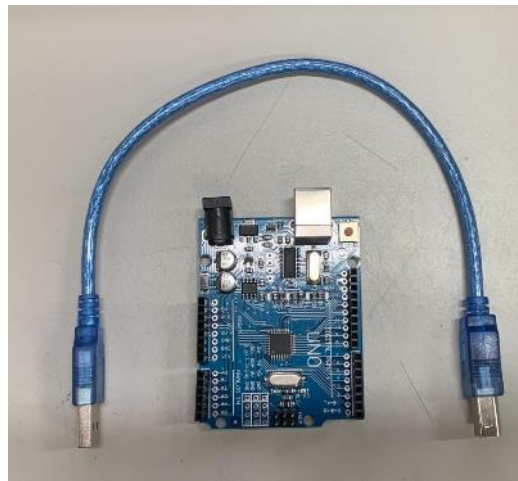
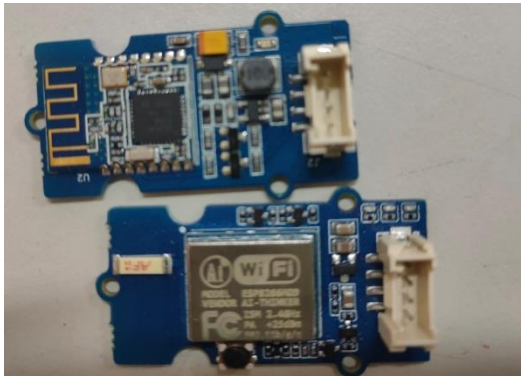
### † 實驗器材認識



Touch	Buzzer	Temperature	LCD+擴充板
3-Axis Accelerometer	Sound	Button	
Step Motor Driver	Rotary Angle	Light	



Step Motor	LED red/blue/green	Pieze Vibration
------------	--------------------	-----------------



BLE	Arduino UNO
WiFi	傳輸線

## † Arduino Uno 硬體介紹

Arduino 開發板品項眾多，各有其特色，如適合入門的 Uno/Uno R3、快閃記憶體 (flash memory) 空間大的 Mega/Mega2560 或是能夠連網的 Yún。由於使用到的 Arduino 開發板為 Uno，如圖 1 所示，以下將以 Uno 為範例，做詳細的硬體介紹。

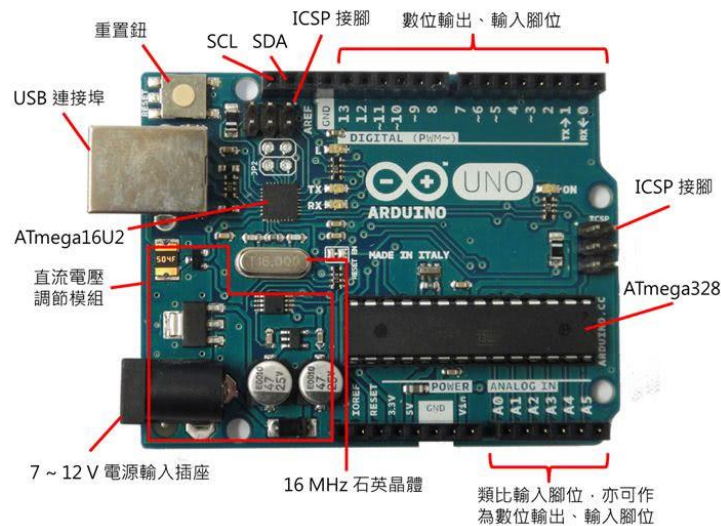


圖 1. Arduino Uno 的腳位圖。

Arduino Uno 是一款基於 ATmega328P 的微控制器板。它有 14 個數位輸入/輸出接腳（其中 6 個可用作 PWM 輸出）、6 個類比輸入、16 MHz 石英晶體、USB 連接孔、電源插孔、ICSP 接頭和重置按鈕，基本規格如表 1 所示。

表 1. Arduino Uno 規格。

名稱	規格
微控制器	ATmega328P
工作電壓	5V
建議輸入電壓	7-12V
限制輸入電壓	6-20V

## † Arduino Uno 腳位介紹

### ● 電源

Arduino Uno 板可通過 USB 連接或外部電源供電。外部（非 USB）電源可以 2 使用電源供應器、9V 電池，通過將 2.1mm 插頭插入電路板的電源插孔供電。注意如果供電電壓低於 7V，則 5V 輸出腳位可能電壓不足，電路板可能會變得不穩定。如果使用 12V 以上，電壓調節器可能會過熱並損壞電路板。開發板上有個標示為 ON 的 LED，用來指示電源供應狀況，如果已提供電源的情況下，發現標示為 ON 的 LED 閃爍、黯淡或熄滅，應儘快拔除電源，檢查電源供應是否正常，以避免損壞控制板。Arduino Uno 上與電源有關的接腳列於表 2 所示。

表 2. 電源腳位介紹

腳位名稱	介紹
VIN	電路板在使用外部電源時的輸入電壓，可以通過此腳位提供電壓。
5V	從電路板上的穩壓器輸出 5V 穩壓電壓。
3.3V	由穩壓器產生的 3.3V 電源。最大電流消耗為 50 mA。
GND	接地。
IOREF	讓其他設備知道 Arduino 控制板的運作電壓。

### ● 類比輸入腳位

A0 至 A5 (A 代表 Analog) 可用來接受類比電壓輸入，但不能輸出類比電壓。每隻腳預設會將 0V 到 5V 轉換為 0 至 1023 的數值。對於輸出電壓為其他範圍的電路模組，可以透過 AREF 腳與函式 `analogReference()`，來提供參考電壓。A0 至 A5 也可作為數位輸出、輸入腳位使用，此時 A0 至 A5 分別可視為 D14 至 D19。

### ● 數位輸出、輸入腳位

D0 到 D13 (D 代表 Digital) 可輸出高電位 5V 與低電位 0V 的數位訊號，也可接受數位訊號。

## † Arduino Uno 數位 IO 腳位設定和輸出控制

在 Arduino Uno 中的 0-13 腳可以自行規劃作數位輸入或數位輸出控制，如圖 2 綠寬所示，程式語法以下。

- `pinMode` (接腳編號, `INPUT` 或 `OUTPUT`)
- `digitalWrite` (接腳編號, `HIGH` 或 `LOW`)

## † 脈波寬度調變 (PWM)

PWM 為 Pulse-Width Modulation 簡寫，中譯脈衝寬度調變，是一種利用透過數位的方式所產生的類比訊號，通過高頻率數位訊號快速切換，使平均電壓因佔空比 (duty cycle) 改變，進而達到模擬類比訊號輸出的效果，可以用來控制燈光亮度、馬達轉速、喇叭頻率或大小聲。

圖 2 中，垂直的綠線代表一個規律的時間週期，在 Arduino 中，預設每個週期是 2ms (PWM 頻率為 500Hz)。`analogWrite()` 的參數值範圍介於 0 到 255 之間，`analogWrite(255)` 代表產生 100% duty cycle 的輸出 (一直通電)，而 `analogWrite(127)` 則是產生 50% duty cycle 的輸出 (一半的時間通電，一半的時間斷電)。



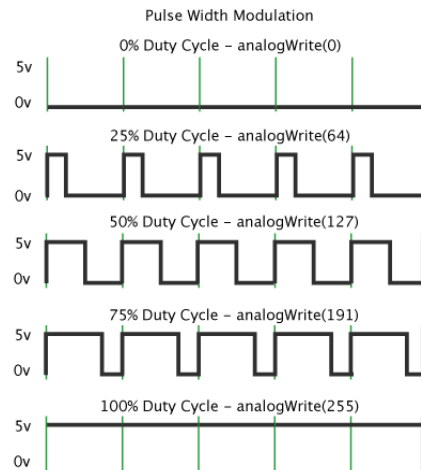


圖 2. Arduino PWM 設定示意圖。

## † Arduino Uno 類比 IO 腳位設定和輸出控制

在 Arduino UNO 中的 3、5、6、9、10、11 號腳可支援 PWM 輸出，其程式語法以下。

- **analogWrite** (接腳編號, 0~255)

在 Arduino UNO 中的 A0、A1、A2、A3、A4、A5 號腳可支援類比輸入控制，如圖 3 所示。

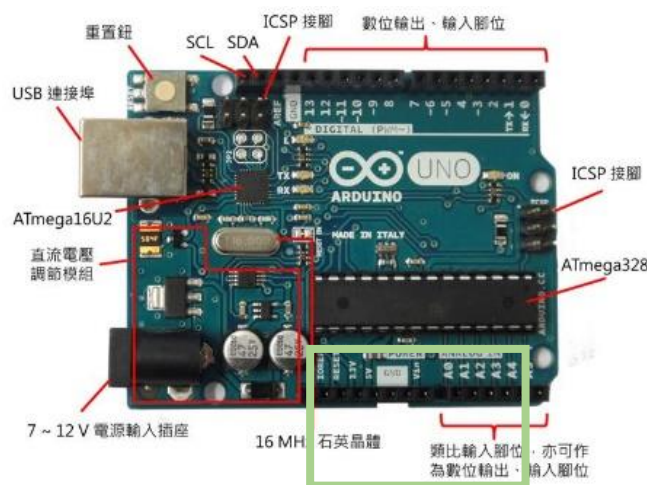


圖 3. 類比輸入腳位示意圖。

使用 Arduino 類比腳位輸入時，程式語法以下

- **analogRead** (接腳編號)
  - 接腳編號：A0~A5
  - 傳回值：0~1023

## † I<sup>2</sup>C 通訊介紹

I<sup>2</sup>C 通訊是用於微控制器和新一代專用積體電路之間的序列資料交換的系統。當它們之間的距離很短時使用它（接收器和發射器通常在同一印刷板上）。通過兩根導線建立連線。一個用於資料傳輸，另一個用於同步（時鐘訊號）。

如下圖 5 所示，一個裝置始終是主裝置。它在通訊開始之前執行一個從晶片的定址。這樣，一個微控制器就可以與 112 個不同的裝置通訊。波特率通常為 100 Kb /秒（標準模式）或 10 Kb/秒（慢波特率模式）。最近出現了波特率為 3.4 Mb/sec 的系統。通過 I<sup>2</sup>C 匯流排通訊的裝置之間的距離限制在幾米。

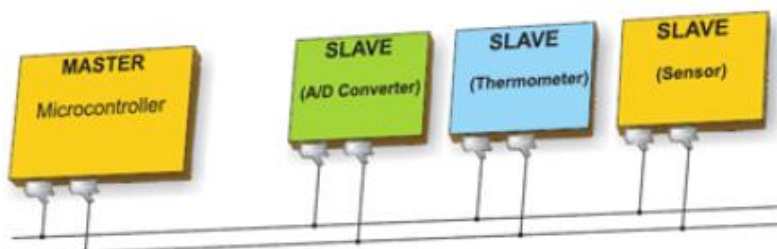


圖 5. I<sup>2</sup>C 通訊介紹示意圖。

## † Arduino 的 I<sup>2</sup>C 通訊接腳

I<sup>2</sup>C 匯流排由兩個訊號組成 - SCL 和 SDA。SCL 是時鐘訊號，SDA 是資料訊號。當前匯流排主控器始終生成時鐘訊號。某些從裝置有時可能會強制時鐘低電平以延遲主裝置傳送更多資料（或者在主裝置嘗試計時之前需要更多時間來準備資料）。這被稱為“時鐘拉伸”。

下圖 6 是不同 Arduino Uno 板的接腳，分別為類比接腳的 A4(SDA)、A5(SCL)和以及在右上接腳的 SDA 和 SCL。



圖 6. Arduino 的 I<sup>2</sup>C 通訊接腳示意圖。

## † LCD

教材中的 LCD 顯示器可顯示 16\*2 個字元，每個字元由 5\*8 的點陣所組成，一般來說都是以 ASCII 碼設定顯示字元，如下表。

表 3. ASCII 碼字元表。

代碼解釋：Dec：10進制 Hx：16進制 Oct：8進制 Char：字元													
16進制表示法：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F、10 ...													
Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!"	65	41	A	97	61	a
^B	2	02		STX	34	22	!"#	66	42	B	98	62	b
^C	3	03		ETX	35	23	!"#\$	67	43	C	99	63	c
^D	4	04		EOT	36	24	!"#\$%	68	44	D	100	64	d
^E	5	05		ENQ	37	25	!"#\$%&	69	45	E	101	65	e
^F	6	06		ACK	38	26	!"#\$%&'	70	46	F	102	66	f
^G	7	07		BEL	39	27	!"#\$%&'(	71	47	G	103	67	g
^H	8	08		BS	40	28	!"#\$%&'()	72	48	H	104	68	h
^I	9	09		HT	41	29	!"#\$%&'()*	73	49	I	105	69	i
^J	10	0A		LF	42	2A	!"#\$%&'()*+	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	!"#\$%&'()*+,	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	!"#\$%&'()*+,-	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	!"#\$%&'()*+,-.	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	!"#\$%&'()*+,-./	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	!"#\$%&'()*+,-./0	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	!"#\$%&'()*+,-./01	80	50	P	112	70	p
^Q	17	11		DC1	49	31	!"#\$%&'()*+,-./012	81	51	Q	113	71	q
^R	18	12		DC2	50	32	!"#\$%&'()*+,-./0123	82	52	R	114	72	r
^S	19	13		DC3	51	33	!"#\$%&'()*+,-./01234	83	53	S	115	73	s
^T	20	14		DC4	52	34	!"#\$%&'()*+,-./012345	84	54	T	116	74	t
^U	21	15		NAK	53	35	!"#\$%&'()*+,-./0123456	85	55	U	117	75	u
^V	22	16		SYN	54	36	!"#\$%&'()*+,-./01234567	86	56	V	118	76	v
^W	23	17		ETB	55	37	!"#\$%&'()*+,-./012345678	87	57	W	119	77	w
^X	24	18		CAN	56	38	!"#\$%&'()*+,-./0123456789	88	58	X	120	78	x
^Y	25	19		EM	57	39	!"#\$%&'()*+,-./0123456789:	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	!"#\$%&'()*+,-./0123456789:;	90	5A	Z	122	7A	z
^[	27	1B		ESC	59	3B	!"#\$%&'()*+,-./0123456789:;,<	91	5B	[	123	7B	{
^\	28	1C		FS	60	3C	!"#\$%&'()*+,-./0123456789:;,<=	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	!"#\$%&'()*+,-./0123456789:;,<=>	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	!"#\$%&'()*+,-./0123456789:;,<=>?	94	5E	^	126	7E	~
^~	31	1F	▼	US	63	3F	!"#\$%&'()*+,-./0123456789:;,<=>?@	95	5F	_	127	7F	~*

## † RFID 模組(RC522)

我們使用到的 RFID 為 Mifare 規格，Mifare 是 NXP（恩智普）半導體公司推出的非接觸型 IC 卡（也就是外表沒有金屬接點的卡片），在市場上獲得廣泛的採用，像是停車場的感應幣（token）、現金卡（如：台灣的悠遊卡）、員工識別證...等等。Mifare 卡有不同的系列，如：Mifare Classic, Mifare UltraLight, Mifare Pro...等，主要的差別在於資料安全加密和驗證的等級。

底下是 Mifare 裝置的概略圖，它的卡片屬於**被動式**、無內建電源（也稱為「無源」），卡片所需的電力來自讀寫器的電磁場。

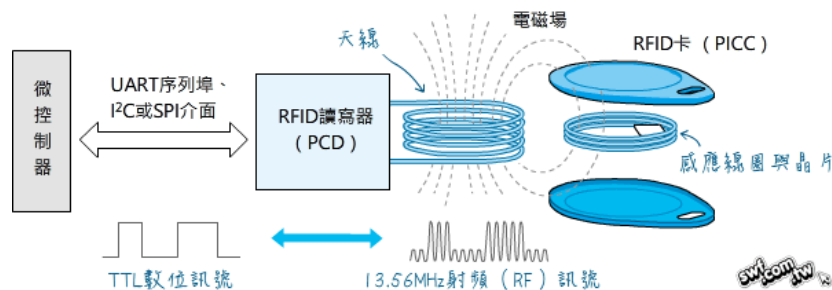


圖 7. RFID 運作方式。

啟動 RFID 讀寫器之後，讀寫器的天線將不停地發送電磁波，每當卡片進入此電磁場，卡片內部的線圈和電路將與此電磁場產生共振，從而獲得電能。無線充電器也是透過電磁感應、非直接連線方式獲取能量，RFID 讀寫器和卡片兩端的線圈（天線）相當於變壓器裡的主線圈和副線圈。

當卡片被電磁能啟動（activate）之後，將等待接收與回應來自讀寫器的命令。微控器和 Mifare 讀寫器之間採用 TTL 數位訊號傳遞資料，為了用電波傳送數位訊號，必須將數位訊號加上載波調變，Mifare 的載波頻率是 13.56MHz。

## †RFID 門禁管理系統實驗步驟

1. 利用 LCD 顯示 RFID 感測到的 UID 碼。

(1) 按照圖 8 圖 9 接線，並將 LCD 連接至 I2C 腳位。

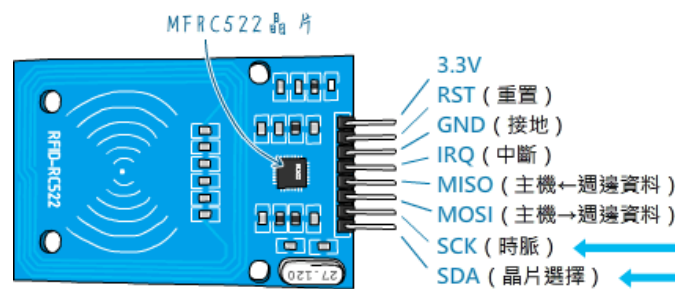


圖 8. RFID 接腳用途。

Signal	MFR522 Reader/PCD Pin	Arduino Uno/101 Pin
RST/Reset	RST	9
SPI SS	SDA(SS)	10
SPI MOSI	MOSI	11 / ICSP-4
SPI MISO	MISO	12 / ICSP-1
SPI SCK	SCK	13 / ICSP-3

圖 9. RFID 接線接腳說明。

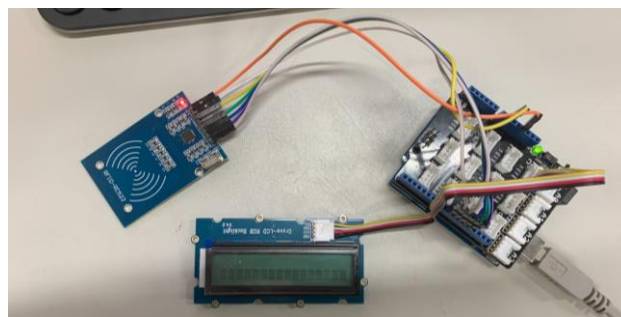


圖 11. 接線示意圖。



(2) 匯入 MFRC522 函式庫，打上以下範例碼。

```
#include<SPI.h>
#include<MFRC522.h>
#include <Wire.h>
#include "rgb_lcd.h"
#define RST_PIN      9
#define SS_PIN       10

MFRC522 mfrc522(SS_PIN, RST_PIN);
rgb_lcd lcd;
char *reference;

byte uid[]={0xF2, 0x57, 0x15, 0x89}; //這是我們指定
的卡片 UID，可由讀取 UID 的程式取得特定卡片的
UID，再修改這行

void setup() {
  Serial.begin(9600);
  Serial.println("RFID reader is ready!");
  lcd.begin(16,2);
  SPI.begin();
  mfrc522.PCD_Init();
  mfrc522.PCD_DumpVersionToSerial();}

void loop() {
  // put your main code here, to run repeatedly:
  if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
    //byte *id = mfrc522.uid.uidByte;    // 取得卡
片的 UID

    Serial.print("id:");
    lcd.write("id=");
    for (byte i = 0; i < 4; i++) { // 逐一顯示
UID 碼

        Serial.print(mfrc522.uid.uidByte[i]);    //
以 16 進位顯示 UID 值

        Serial.print(" ");
        lcd.print(mfrc522.uid.uidByte[i]);
        lcd.write(" ");
    }
    Serial.println();
    mfrc522.PICC_HaltA(); // 讓卡片進入停止模
式
    delay(1000);
    lcd.clear();
  }
}
```

(3) 編譯成功後上傳至開發板，拿著設定成會員 RFID 卡掃感應板，即可得到下圖資訊。

```
18:16:39.767 -> Card UID: F2 57 15 89
18:16:39.767 -> PICC type: MIFARE 1KB
18:16:39.802 -> Access Granted!
```

圖 12. RFID 測試碼執行成功之序列埠畫面。

```
RFID reader is ready!
Firmware Version: 0x11 = (unknown)
id:37 46 102 103
member1
```



圖 13.序列埠與 LCD 畫面。

以下為範例碼，將\*\*填滿。

```
//匯入所需函式庫
#include<SPI.h>
#include<MFRC522.h>
#include <Wire.h>
#include "rgb_lcd.h"
//設定 PIN 腳
#define RST_PIN      **
#define SS_PIN       **
// 宣告 RFID 讀取物件
MFRC522 mfrc522(SS_PIN, RST_PIN);
// 宣告 LCD 物件
rgb_lcd lcd;

// 會員結構體
struct Member {
    String cardNumber;
    String name;
};

// 設定會員列表
const int number = 1; // 會員數量
Member members[number] = {
{"XXXXXXX",Member1}, // 將"XXXXXXX" 替換為
會員 1 的實際 RFID 卡號
};

void setup() {
    Serial.begin(9600);
    Serial.println("RFID reader is ready!");
    /**/lcd 初始化
    /**/SPI 初始化
    /**/MFRC522 初始化
    mfrc522.PCD_DumpVersionToSerial();
}

void loop() {
    //檢測 RFID 卡
    if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
        // 讀取 RFID 卡號
```

```
String cardNumber = "";
for (byte i = 0; i < mfrc522.uid.size; i++) {
    cardNumber += String(mfrc522.uid.uidByte[i] <
0x10 ? "0" : "");
    cardNumber += String(mfrc522.uid.uidByte[i],
HEX);
}
Serial.println("Card Number: " + cardNumber);

// 比對 RFID 卡號
bool memberFound = false;
for (int i = 0; i < number; i++) {
    if (cardNumber == members[i].cardNumber) {
        Serial.println("Access Granted - Welcome, " +
members[i].name);
        lcd.clear();
        lcd.print("Welcome, ");
        lcd.print(members[i].name);
        memberFound = true;
        break;
    }
}

// 如果不是會員
if (!memberFound) {
    *****
}

delay(1000); // 延遲一秒避免連續讀取
lcd.clear();
lcd.print("Waiting for card");
}}
```

## †實驗步驟

### 2. 利用 LED 燈先確認 Arduino 運作正常

(1) 將擴充板與 Arduino 連接起來，如下圖所示。



圖 14. 擴充板與 Arduino 連接示意圖。

(2) 將 Arduino 插上 USB 線與電腦連接。

(3) 點選開發介面的工具欄確認開發板為”Arduino Uno”，確認序列埠為正確位置(開啟裝置管理員查看)。

注意：序號列可能不會顯示 Arduino Uno，可以藉由插入後，新增序號列確認 Arduino Uno 的位置。

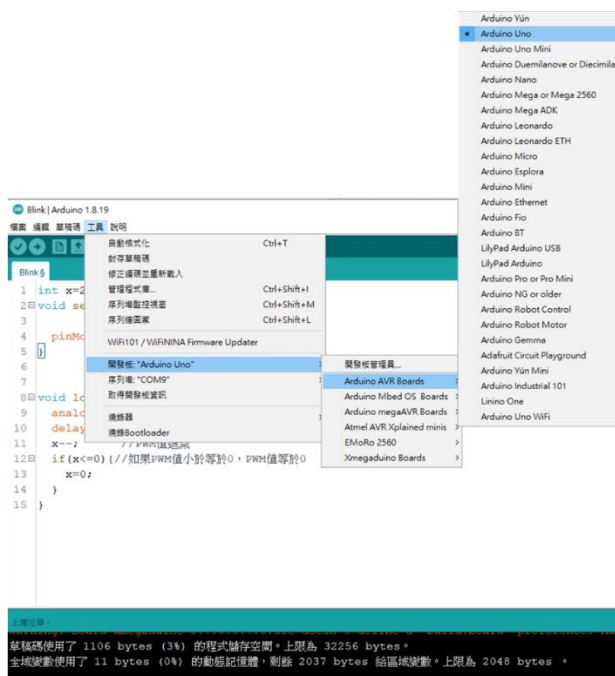


圖 15. 開發板與序列埠確認示意圖。

(4) 將 LED 連接至擴充板 D4。



圖 16. 感測器連接示意圖。

(5) 由檔案/範例開啟測試檔 Blink。

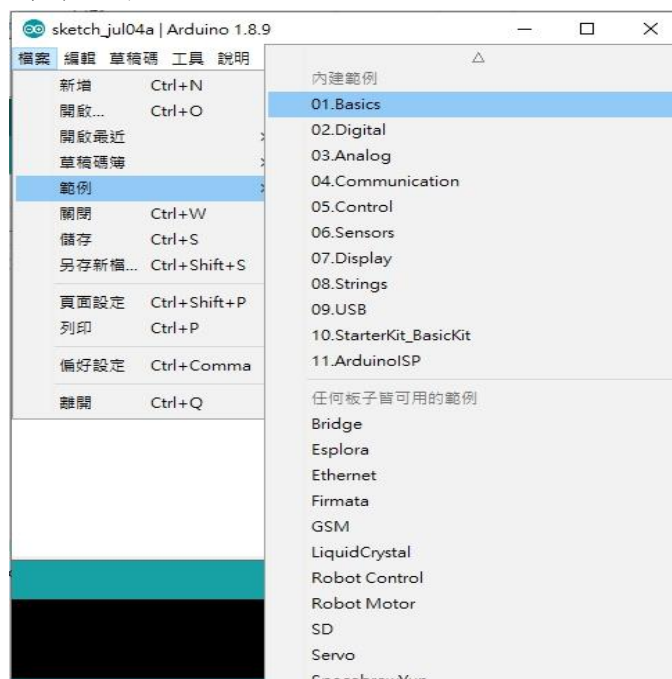


圖 17. 測試檔開啟位置示意圖。



(6) 紅框處改為 4。

```
void setup() {  
  // initialize digital pin  
  pinMode(4, OUTPUT);  
}  
  
// the loop function runs ov  
void loop() {  
  digitalWrite(4, HIGH);  
  delay(1000);  
}
```

\*setup()設定腳位用途，loop()編寫主程式。

(7) 編譯成功後上傳至開發板，即可看到 LED 閃爍，若沒閃爍請檢察序列埠是否正確，或是更改 USB 電源設定(控制台>硬體和音效>編輯電源計劃>變更進階電源設定>USB 設定停用)。



圖 18. 編譯成功示意圖。

3. 設計防彈跳延遲(debounce delay)，使按鈕每按一下便改變一次狀態。

(1) 如下圖 19 連接 LED 與按鈕(或觸摸感測器)。

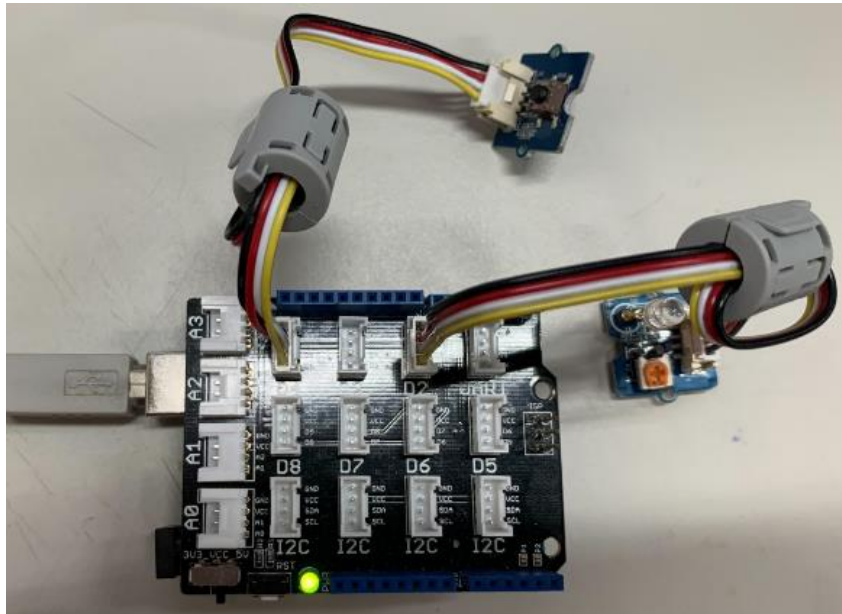


圖 19. 接線示意圖。

(2) 參考以下示範碼編寫程式碼。

```
#define buttonPin 4 //按鈕腳位
#define ledPin 2 //LED腳位

int ledState = HIGH; // 儲存LED狀態
int buttonState; // 儲存按鈕狀態
int lastButtonState = LOW; // 儲存上一次按鈕狀態

unsigned long lastDebounceTime = 0; // 紀錄 millis()
unsigned long debounceDelay = 50; // 過50ms才能轉換狀態

void setup() {
  pinMode(buttonPin, INPUT); //設定按鈕為輸入
  pinMode(ledPin, OUTPUT); //設定LED為輸出
  digitalWrite(ledPin, ledState); //設定LED初始狀態
}

void loop() {
  int reading = digitalRead(buttonPin);

  if (reading != lastButtonState) {
    lastDebounceTime = millis(); //按鈕狀態改變，重置時間基準
  }

  if ((millis() - lastDebounceTime) > debounceDelay) { //過了50ms後

    if (reading != buttonState) {
      buttonState = reading;

      if (buttonState == HIGH) { //按下按鈕時
        ledState = !ledState; //改變燈號
      }
    }
  }

  digitalWrite(ledPin, ledState); //控制LED燈號

  lastButtonState = reading; //儲存按鈕狀態
}
```

(3) 編譯並上傳，程式效果為按鈕每按一下就會改變燈號。

#### 4. LCD 顯示器功能學習

(1) 將 LCD 連接至 I2C 腳位。

(2) 草稿碼→匯入程式庫→加入 ZIP 程式庫，找到 Grove\_LCD\_RGB\_Backlight-master.zip，

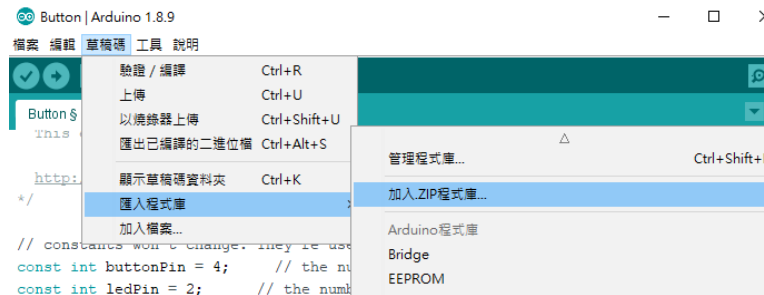


圖 20. 新增程式庫路徑圖。

(3) 執行此函式庫附屬的範例碼，藉由觀察 LCD 的呈現效果瞭解外接程式庫中各個程式的用途(如 lcd.begin()、lcd.write()、lcd.print()、lcd.setCursor()、lcd.clear() 等等程式碼的用途及效果)，範例碼由下圖路徑開啟。

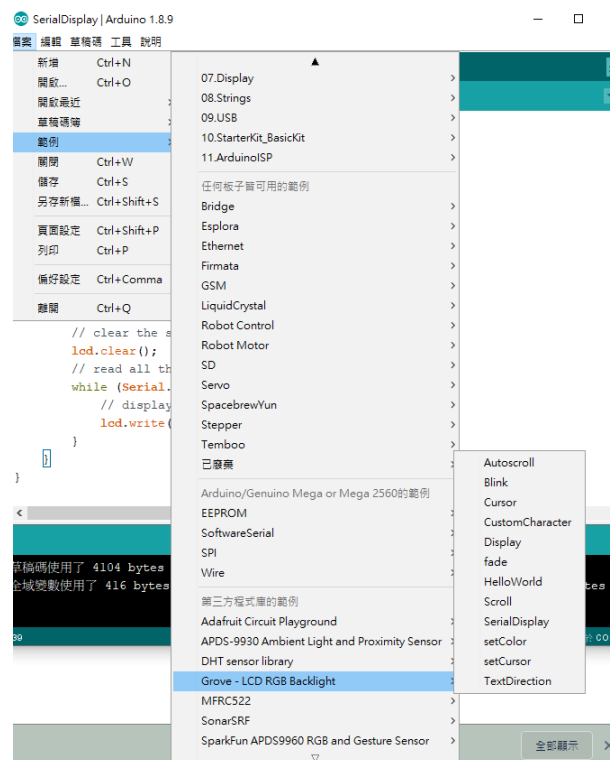


圖 21. 範例碼路徑圖。

#### 5. 蜂鳴器模組功能練習

(1) 將蜂鳴器模組連接在 PWM 的輸出腳，如圖 15 所示。

```

1 int x=255;
2 void setup() {
3
4     pinMode(6, OUTPUT); //設定PWM輸出腳
5 }
6
7
8 void loop() {
9     analogWrite(6,x); //PWM輸出，初值255
10    delay(100); //延遲時間100ms
11    x--; //PWM值遞減
12    if(x<=0){ //如果PWM值小於等於0，PWM值等於0
13        x=0;
14    }
15 }

```

圖 22. 範例碼示意圖。

- (2) 參考以上示範碼編寫程式碼。
- (3) 編譯並上傳，程式效果每 100ms 蜂鳴器聲音降低。

◆自己動手做：完成簡易門禁管理系統。

- (1) 設定會員名單(2 個或以上，利用悠遊卡或學生證掃感應器，並記錄下 UID。
- (2) 讓序列埠視窗顯示如左，LCD 顯示如右

**做完主動找助教檢查喔！**

◆作業 1：完成程式碼

- 功能:修改自己動手做，是會員時背景顯示藍色，不是會員時顯示紅色，並且加入蜂鳴器，是會員時鳴叫短音，不是會員時鳴叫長音。

◆作業 2：回答問題

1. 試說明 debounce delay 的功能，以及與 delay()差別是什麼。
2. 解釋 lcd.begin()、lcd.write()、lcd.print()、lcd.setCursor()、lcd.clear() 的用途以及括號內寫法，並說明 lcd.write()、lcd.print() 的區別。
3. 舉例說明 pinMode()、digitalRead()、digitalWrite()、analogWrite()、delay() 用法，() 內該寫什麼。  
(如:pinMode() 可以設定腳位為輸入或輸出，pinMode(5,OUTPUT) 即為將 D5 腳位設為輸出，可以用於 LED 燈控制。)
4. 分別說明 MFRC522.PCD\_Init()、MFRC522.PICC\_ReadCardSerial()、MFRC522.PCD\_DumpVersionToSerial()、MFRC522.uid.uidByte()MFRC522.PICC\_IsNewCardPresent()、的意義與用法。
5. 承 1，若要使用 MFRC522 函式庫，應該要在程式的最上方以及 setup() 中加哪些程式碼(總共 5 行)。